PCクラスタでの多倍長計算の実装と利用

藤原, 宏志
京都大学大学院情報学研究科

https://doi.org/10.15017/1470174

出版情報：九州大学情報基盤研究開発センター全国共同利用システム広報. 2 (3), pp.93-96, 2009-03.
九州大学情報統括本部広報委員会
バージョン：
権利関係：

PC クラスタでの多倍長計算の実装と利用

藤原宏志*

概 要

本研究は，計算機支援の立場から，偏微分方程式や積分方程式の数値計算のための高速多倍長計算環境の設計と実装を目的とするものである．数値的不安定性を有する問題に対して，その計算過程における計算誤差の影響を排除するために，高精度離散化手法と多倍長計算を利用することが高精度なシミュレーションの実現に有効であることが指摘されている．この数値計算手法を計算力学などの大規模問題に適用するために，64 ビットアーキテクチャのプロセサを利用した PC クラスタでの利用を想定した高速多倍長数値計算環境について述べる．また，九州大学のスーパーコンピュータシステムでの利用についても紹介する．

1 緒言

計算力学などの数値シミュレーションでは，実数を扱うことが不可欠である．現在使われている一般的な計算機アーキテクチャでは，実数の表現と演算は浮動小数点数演算 (floating-point arithmetic) [1, 2] によって扱われる．現在の計算機アーキテクチャで実装されている浮動小数点数を Table 1 に示す．これら浮動小数点演算では，二進展開で有限桁の有理数による実数の近似，すなわち表現と演算がおこなわれ，真値との間に誤差が発生する．この誤差を丸め誤差とよぶ．

Table 1: Binary floating-point types

type	format width in bits	the number of significant bits* (precision)	significant precision in decimal	Exponent width in bits	range
IEEE754 single	32	24	7.2	8	$10^{\pm 38}$
IEEE754 double	64	53	16.0	11	$10^{\pm 308}$
IEEE P754 binary128 [3]	128	113	34.0	15	$10^{\pm 4932}$
IA32 extended	80	65	19.6	15	$10^{\pm 4932}$
SPARC extended	128	113	34.0	15	$10^{\pm 4932}$
POWER extended	128	$\geq 106^{\dagger}$	≥ 31.9	11	$10^{\pm 308}$
NVIDIA fp16,ATI fp16 [4]	16	11	3.6	5	$10^{-5} - 10^{5}, 10^{6}$
ATI fp24	24	16	5.1	6	$10^{\pm 19}$

(*) including the hidden one

(†) POWER extended is sum of two IEEE754 doubles

*京都大学大学院 情報学研究科, E-mail: fujiwara@acs.i.kyoto-u.ac.jp

通常の偏微分方程式の初期値問題や境界値問題の設定は順問題と呼ばれ，Hadamard の意味で適切 (well-posed) と呼ばれる性質を満たす設定のもとで扱われることが多い．これは，解が一意に存在し，適当なノルムで安定性を有することを意味する．特に安定性は，初期値や付随するデータの摂動が小さければ解の摂動も小さいことを意味する．数値計算において丸め誤差は摂動に相当するため，丸め誤差を含む環境での数値計算においては，計算対象とする系が安定性を有することが望まれる．すなわち，安定性を有する系の数値計算では，丸め誤差が十分小さければ，丸め誤差が数値解に及ぼす影響は無視できるほど小さいことが期待される．この方針に基づく順問題の数値計算は，計算機の性能向上によって大規模数値計算が実現されたことにより成功を収め，そこでは丸め誤差そのものが中心的話題となることは少なかった．

一方，近年，非破壊検査やリモートセンシングなどの逆問題と呼ばれる問題の扱いが重要となっている．逆問題は，上述の順問題とは著しく性質を異にし，Hadamard の意味で非適切 (ill-posed) と呼ばれる．数値計算の視点からは，系が安定性を有さないことにより，計算過程において丸め誤差が急激に増大して数値計算が破綻する．そのため，従来手法では数値解の構成は不十分である．また，系の不安定性が本質的に重要である場合には，不安定性を直接扱い得る新たな手法の開発が望まれていた．また，数学的にある位相で安定であっても，実際の数値計算において丸め誤差の影響が現れる場合もある．

著者らを含む研究グループは，これら数値的に不安定な問題に対し，高精度離散化手法と多倍長計算の併用による取り扱いを提案している [5, 6]．多倍長計算を利用することで，計算結果に丸め誤差の影響が現れないだけの十分な精度を確保することが可能となり，これによって仮想的に丸め誤差のない数値計算が実現される．従来にも多倍長計算のためのソフトウエアは幾つか提供されているが [7]，計算力学などでの利用は想定されていない場合が多い．そこで本研究では，計算力学などの科学技術計算の利用を想定し，新たな設計と実装をおこなった [8]．

2　多倍長計算環境の設計と実装

提案する多倍長計算環境では，64 ビット符号なし整数の配列によって多倍長精度浮動小数点数を表す．データ構造を Fig. 1 に示す．Fig. 1 により正規化数を表す場合，$b = 2^{62} - 1$ をバイアスとして次の値を表す．

$$(-1)^s \times 2^{e_b - b} \times \left(1 + \sum_{i=1}^{n} f_i 2^{-64i}\right).$$

多倍長数の演算は，高速性を実現するため，対象アーキテクチャを限定してアセンブリ言語で実装をおこなっている．特に偏微分方程式や積分方程式の数値計算において，10 進法で 100 桁から数千桁の精度での高速計算を目的とし，それに適するアルゴリズムを採用している [2]．これまでに

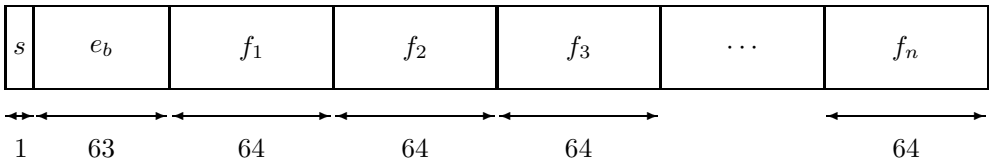

Fig. 1: Format of the proposed floating point multiple-precision number

AMD64, EM64T, IA32, SPARC V9 において実装をおこない，Linux, Solaris, Mac OSX (64bit), Windows (32bit) 上で利用可能である．

現在，四則演算，組み込み関数，10 進法での入出力の機能を備えており，FORTRAN90 および C++ 言語で利用可能である．FORTRAN90 での利用例を Fig. 2 に示す．FORTRAN90 および C++ 言語での利用においては，演算子多重定義などの機能をもちいて組み込みの浮動小数点型と同様の式の記述が可能となっている．

九州大学情報基開発盤センターに設置されているスーパーコンピュータシステム tatara において，FUJITSU PRIMERGY RX200S3 用のプログラムのコンパイル例を Fig. 3 に示す [9]．ファイル `sample.f90` はユーザプログラムであり，`libexfloat.a` は演算を実装するライブラリファイル，`exflib.F90` では多倍長数をモジュールとして定義し，インターフェースを提供する．Fig. 3 では，計算精度を 10 進 100 桁としてコンパイル時に指定している．

```
PROGRAM sample

    USE exflib
    TYPE(exfloat) :: x, y, z

    ! output the current computation precision
    WRITE(*,*) 'Current prec :', exflib_exfloat_precision, 'digits.'

    x = '2 * #PI'
    y = '3.14'
    z = x * y
    z = 2 * x
    z = sin(y)

    ! output with 100 decimal digits
    WRITE(*,*) TRIM(exflib_format('F.100', z))

END PROGRAM sample
```

Fig. 2: Sample program 'sample.f90' in FORTRAN90

```
% frt -pg -Free -Kfast \
  -DEXFLIB_INTERFACE=77 -DEXFLIB_EXFLOAT_PRECISION10=100 \
  -Am exflib.F90 sample.f90 libexfloat.a
```

Fig. 3: Compile with FORTRAN90 on the supercomputer system 'tatara' at Kyushu University

3　多倍長並列計算

多倍長計算においては，計算時間と要求メモリ量が問題となる．提案するライブラリは，OpenMP, MPI による並列計算に対応しており，連立一次方程式の求解 [10] および固有値分解 [11] の並列ソルバの設計と実装をおこなっている．並列計算の例として，PRIMERGY RX200S3 (Xeon 3.0 GHz,

Table 2: Computational time of exflib, MPI parallel LU decomposition on the supercomputer system FUJITSU PRIMERGY RX200S3 at Kyusyu University (unit: sec.)

decimal digits	100	100	100	100	500	500	1000	1000
matrix size	2000	4000	8000	12000	2000	4000	2000	4000
total memory	213M	854M	3.3G	7.5G	824M	3.2G	1.6G	6.3G
4 proc	178	1415	NA	NA	997	NA	2819	NA
16 proc	48	362	2859	NA	256	2016	725	NA
32 proc	26	186	1448	NA	132	1030	372	2910
64 proc	22	104	748	2473	71	534	204	1508

デュアルコア，2 プロセッサ, ジョブクラス cdbg64) クラスタ上での C++ 言語と MPI (mpiFCC) による並列 LU 分解の計算時間の例を Table 2 に示す．これより，計算時間の削減に並列計算が有効であることがわかる．

謝辞　本稿の執筆の機会を与えて頂きました九州大学の渡部善隆先生に，御礼申し上げます．

References

[1] IEEE standard for binary floating-point arithmetic : ANSI/IEEE std 754-1985 (1985). Reprinted in *SIGPLAN* **22** (1987), 9–25.

[2] Knuth, D. E.: *The Art of Computer Programming Volume 2 : Semi Numerical Algorithms*, 3rd ed. Addison-Wesley (1998).

[3] DRAFT Standard for Floating-Point Arithmetic P754, Draft 1.5.0, October 5, 2007.

[4] Pharr, M. ed.: *GPU Gems 2*, Addison-Wesley (2005).

[5] Imai, H. and Takeuchi, T.: Some advanced applications of the spectral collocation method, *GAKUTO Internat. Ser. Math. Sci. Appl.* **17** (2002), 323–335.

[6] Fujiwara, H., Imai, H.,Takeuchi, T. and Iso, Y.: Numerical treatment of analytic continuation on multiple-precision arithmetic, *Hokkaido Math. J.*, **36(4)** (2007), 837–847.

[7] Smith, D. M.: A Fortran package for floating point multiple-precision arithmetic, *Transactions on Mathematical Software* **17** (1991), 273–283.

[8] http://www-an.acs.i.kyoto-u.ac.jp/~fujiwara/exflib.

[9] http://www.cc.kyushu-u.ac.jp/scp/system/general/PQ/super2007.html

[10] 藤原宏志：非適切問題を念頭においた多倍長精度並列 LU 分解について，第 57 回理論応用力学講演会 講演論文集 (2008) 521–522.

[11] 藤原宏志：多倍長計算による固有値分解の実現と応用, 情報処理学会 ハイパフォーマンスコンピューティング研究報告 No.111 (2007-HPC-111), 151–156.